

FIG. 1

→ DATA TRANSFERRED
----> DATA NOT TRANSFERRED

FIG. 2

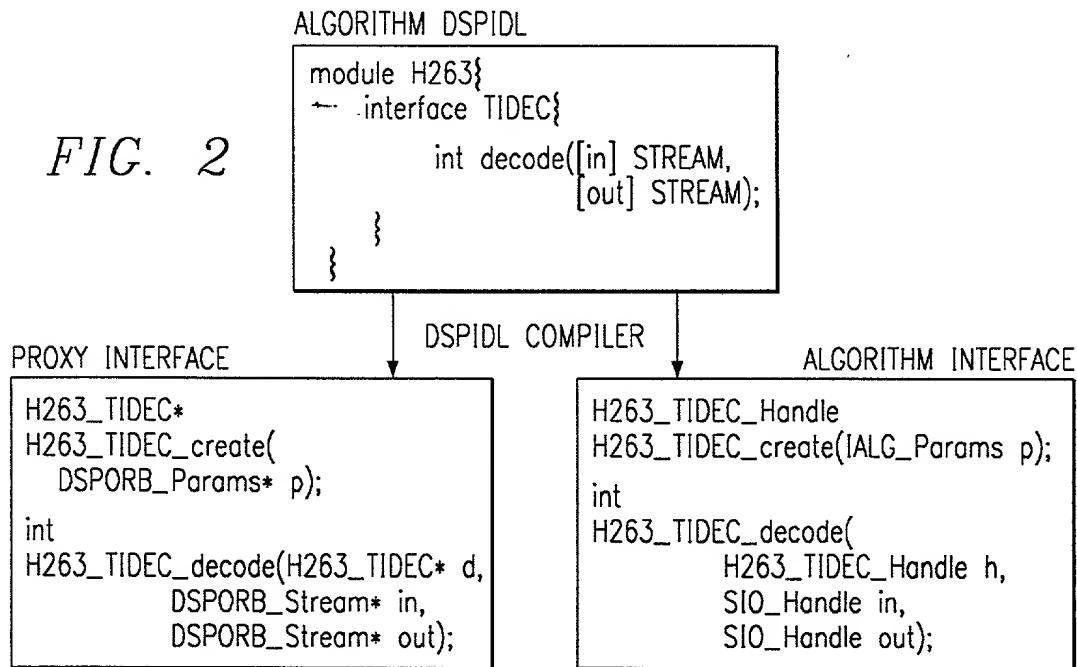
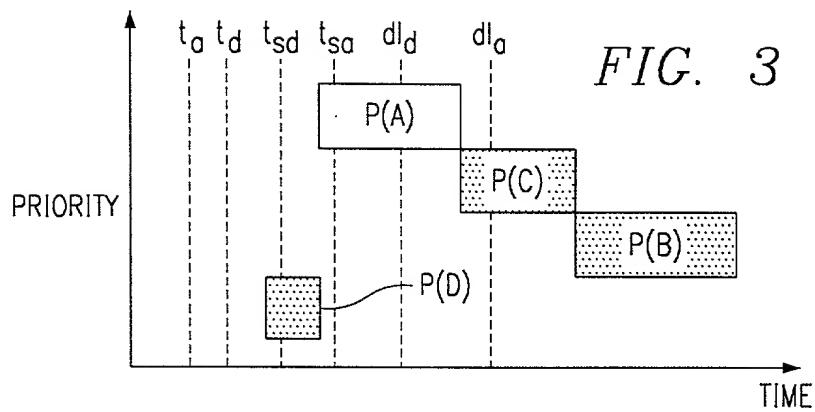


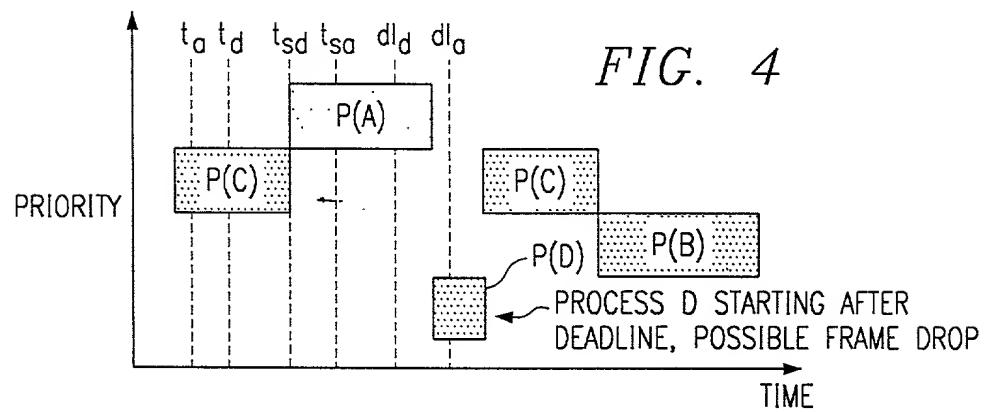
FIG. 3



t_{sa} = LAST POSSIBLE TIME FOR PROCESS A
TO START AND STILL MAKES ITS DEADLINE

t_{sd} = LAST POSSIBLE TIME FOR PROCESS D
TO START AND STILL MAKE ITS DEADLINE

FIG. 4



t_{sa} = LAST POSSIBLE TIME FOR PROCESS A
TO START AND STILL MAKES ITS DEADLINE

t_{sd} = LAST POSSIBLE TIME FOR PROCESS D
TO START AND STILL MAKE ITS DEADLINE

FIG. 5

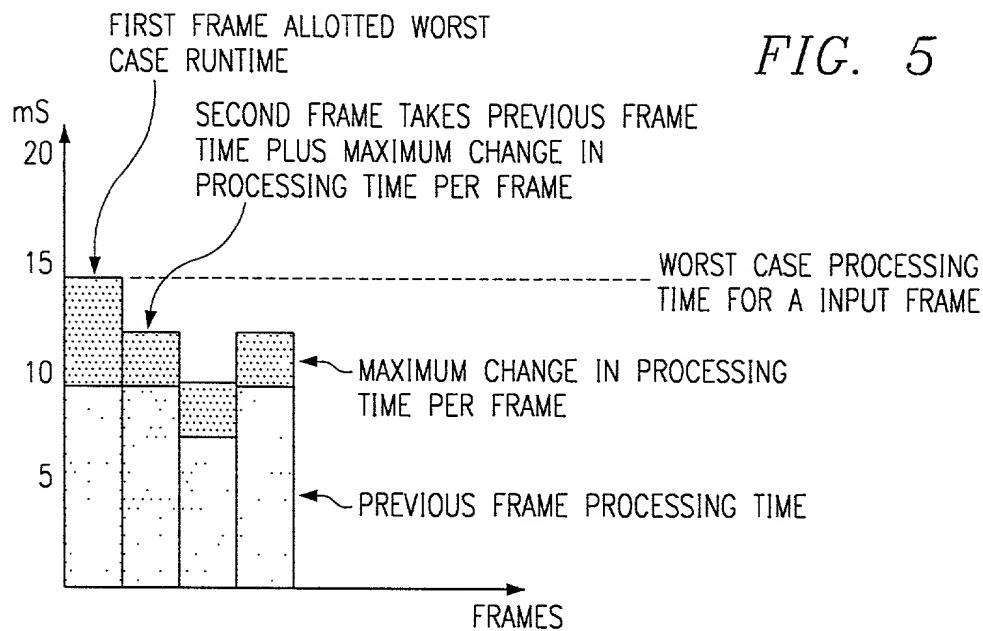


FIG. 6

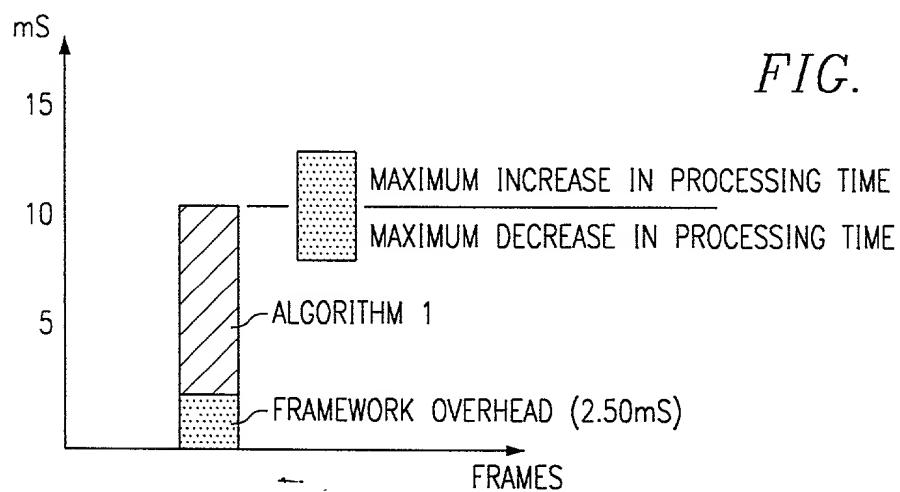


FIG. 7

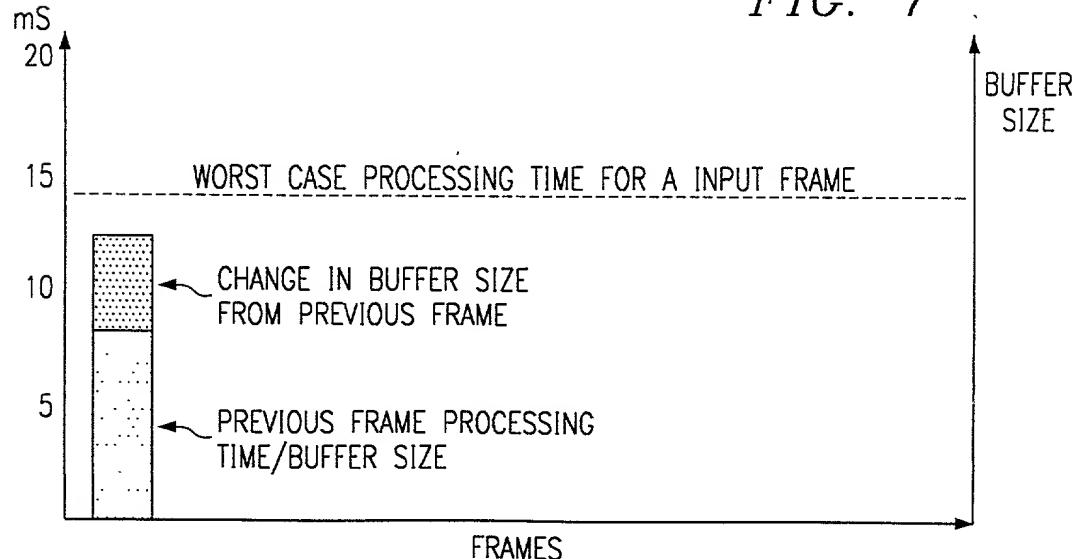


FIG. 8

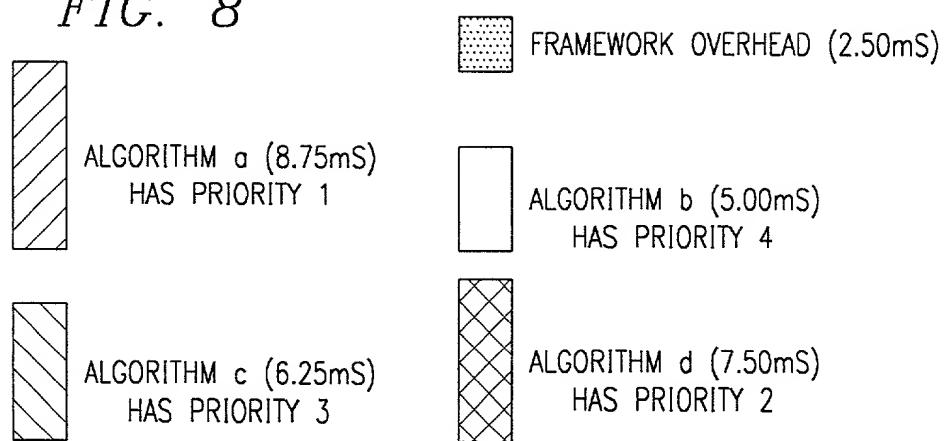
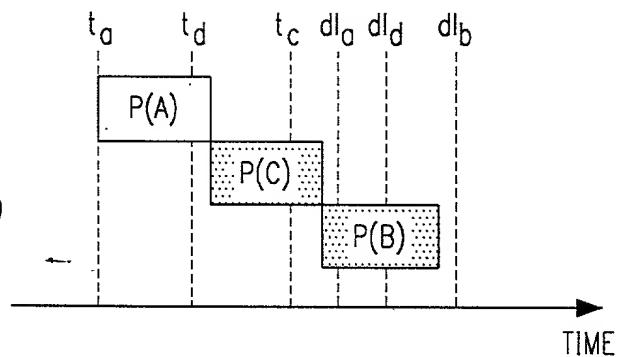


FIG. 9

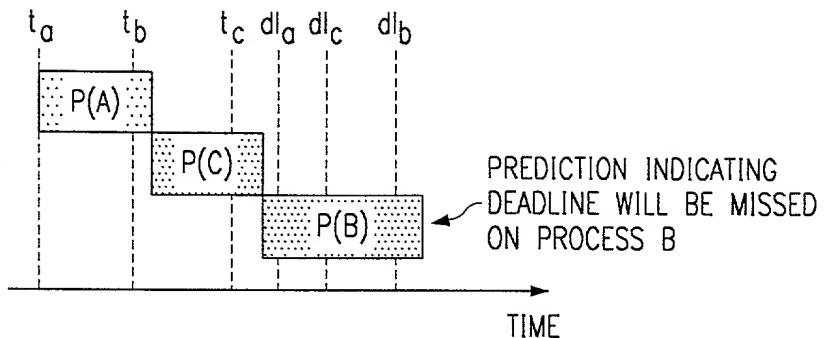


t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS

d_{l_i} = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME

$P()$ = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

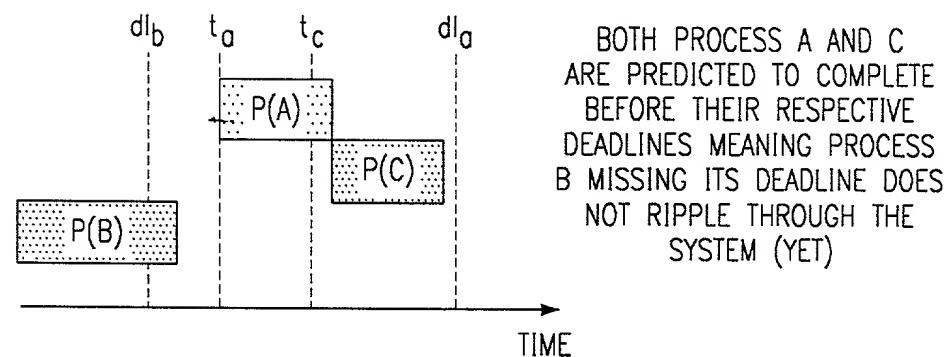
FIG. 10



t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS

dl_i = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME

$P()$ = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME



t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS

dl_i = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME

$P()$ = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

FIG. 11

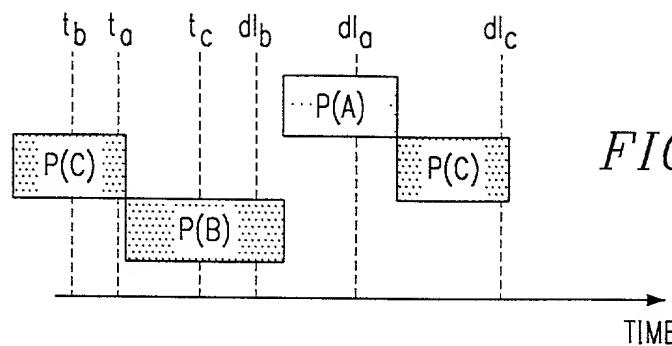


FIG. 12

- t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS
- dl_i = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME
- $P()$ = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

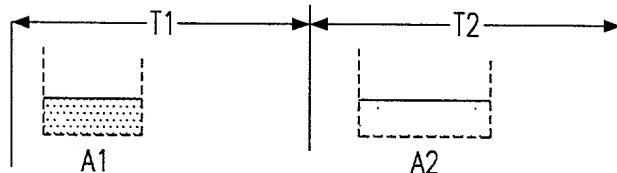


FIG. 13a

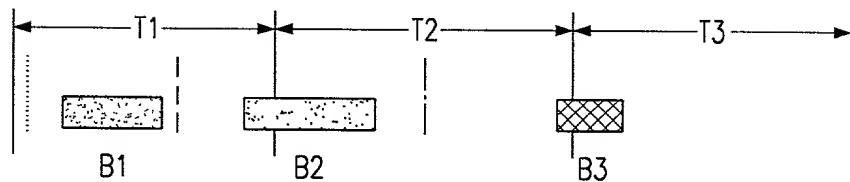


FIG. 13b

..... ARRIVAL OF BUFFER B1
 - - - ARRIVAL OF BUFFER B2
 - - - ARRIVAL OF BUFFER B3

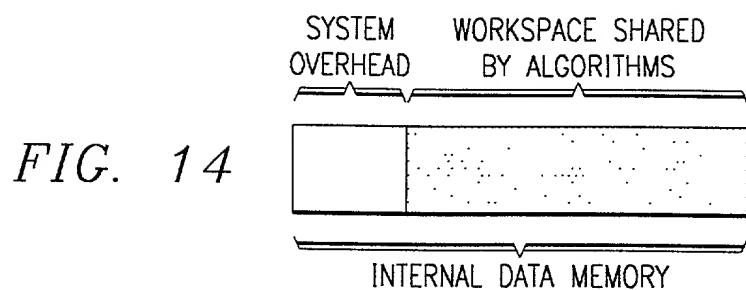


FIG. 14

FIG. 15

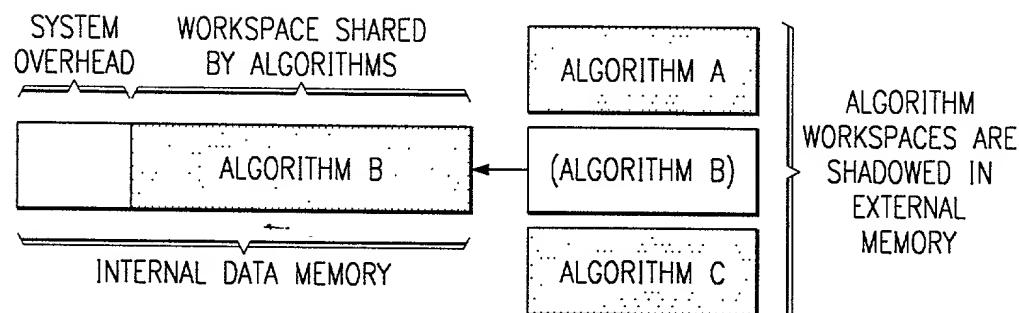
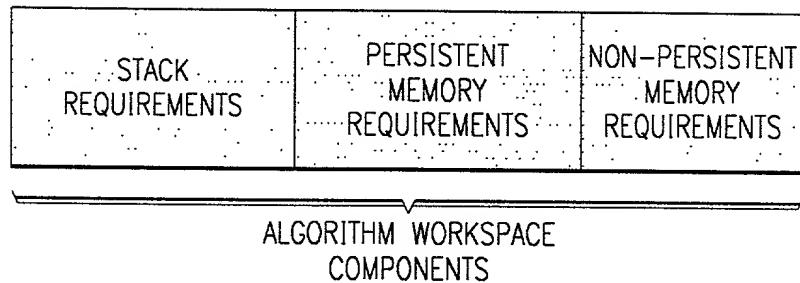
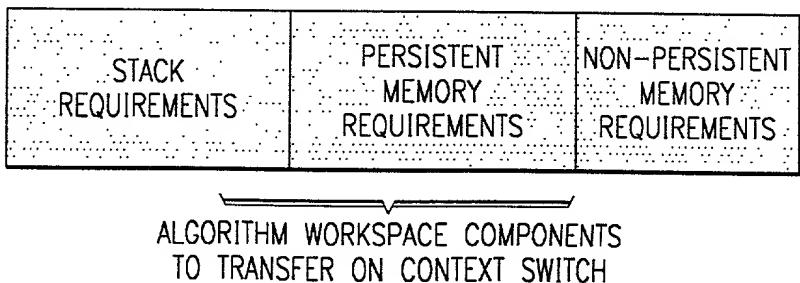


FIG. 16

FIG. 17



STACK REQUIREMENTS	PERSISTENT MEMORY REQUIREMENTS	PERSISTENT READ ONLY MEMORY REQUIREMENTS	NON-PERSISTENT MEMORY REQUIREMENTS
--------------------	--------------------------------	--	------------------------------------

ALGORITHM WORKSPACE COMPONENTS TO TRANSFER IN PRIOR TO ALGORITHM EXECUTION
IF ALGORITHM REQUIRES CONSTANT TABLES
(CONTEXT SWITCH IN ONLY)

FIG. 18

STACK REQUIREMENTS	PERSISTENT MEMORY REQUIREMENTS	PERSISTENT READ ONLY MEMORY REQUIREMENTS	NON-PERSISTENT MEMORY REQUIREMENTS
--------------------	--------------------------------	--	------------------------------------

READ ONLY PERSISTENT MEMORY DOES NOT NEED TO BE TRANSFERRED OUT ON CONTEXT SWITCH. THEREFORE ALGORITHM PAGE CHANGE-OUT IS MORE EFFICIENT.

FIG. 19

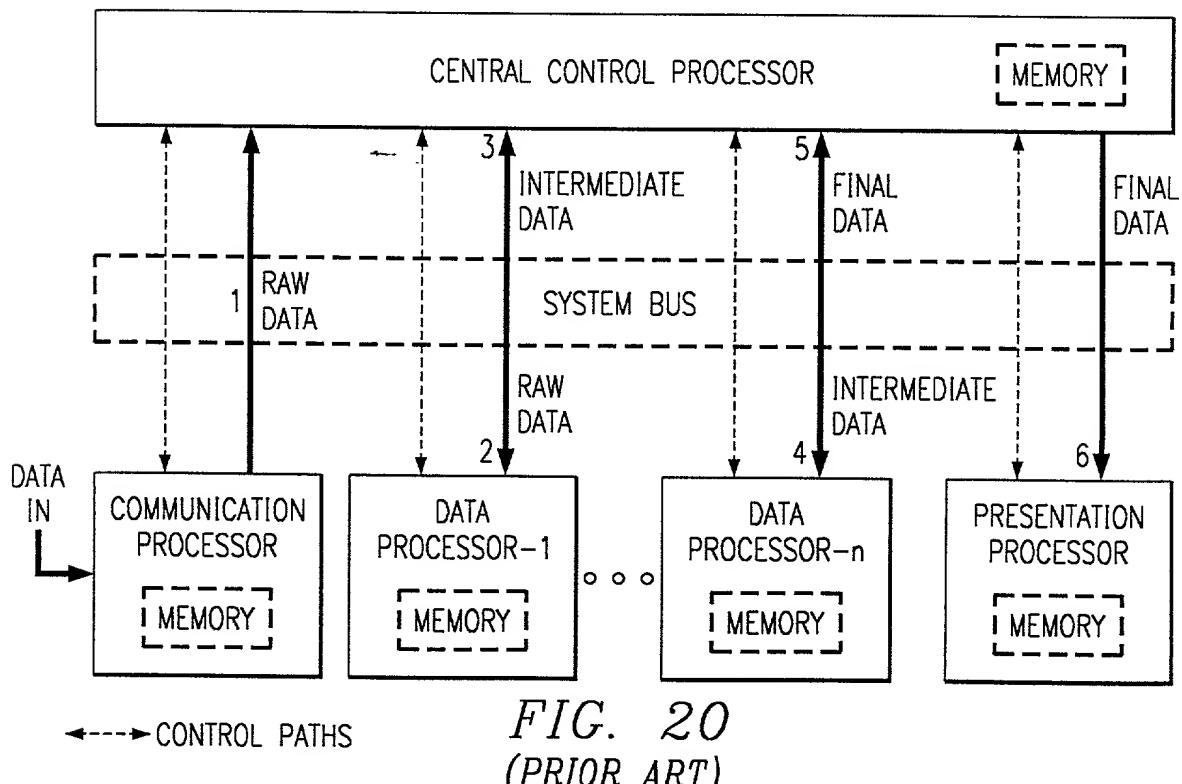


FIG. 20
(PRIOR ART)

FIG. 21

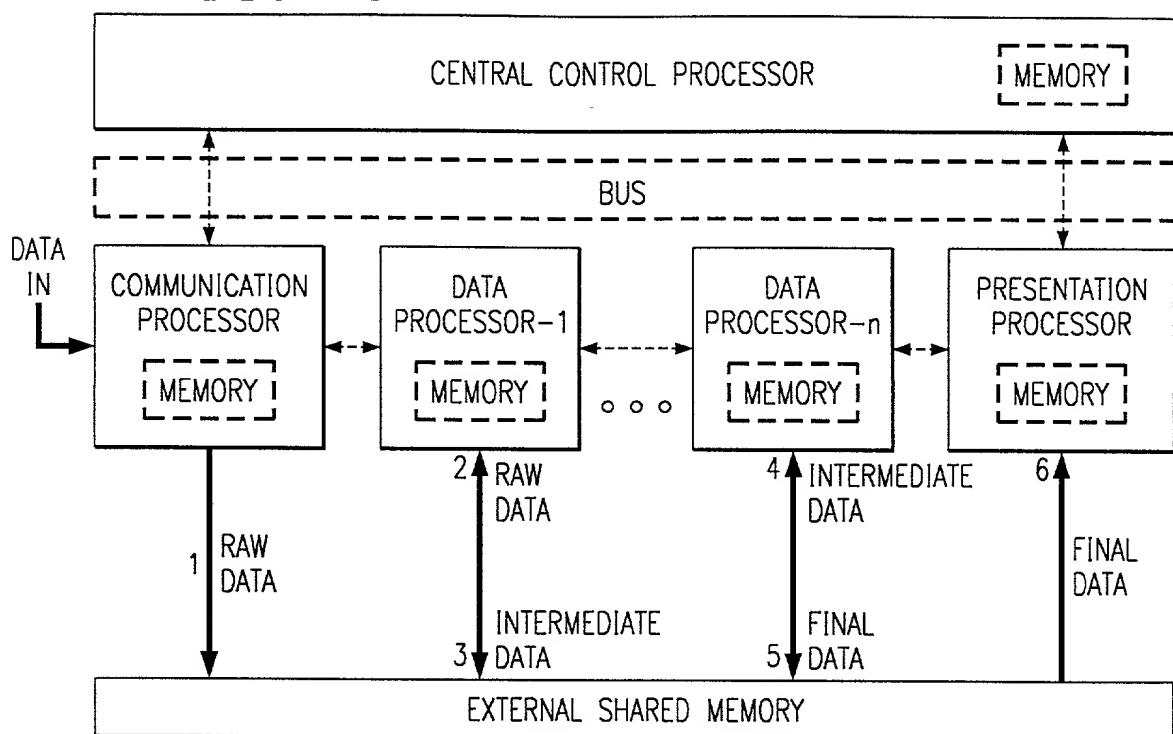


FIG. 22

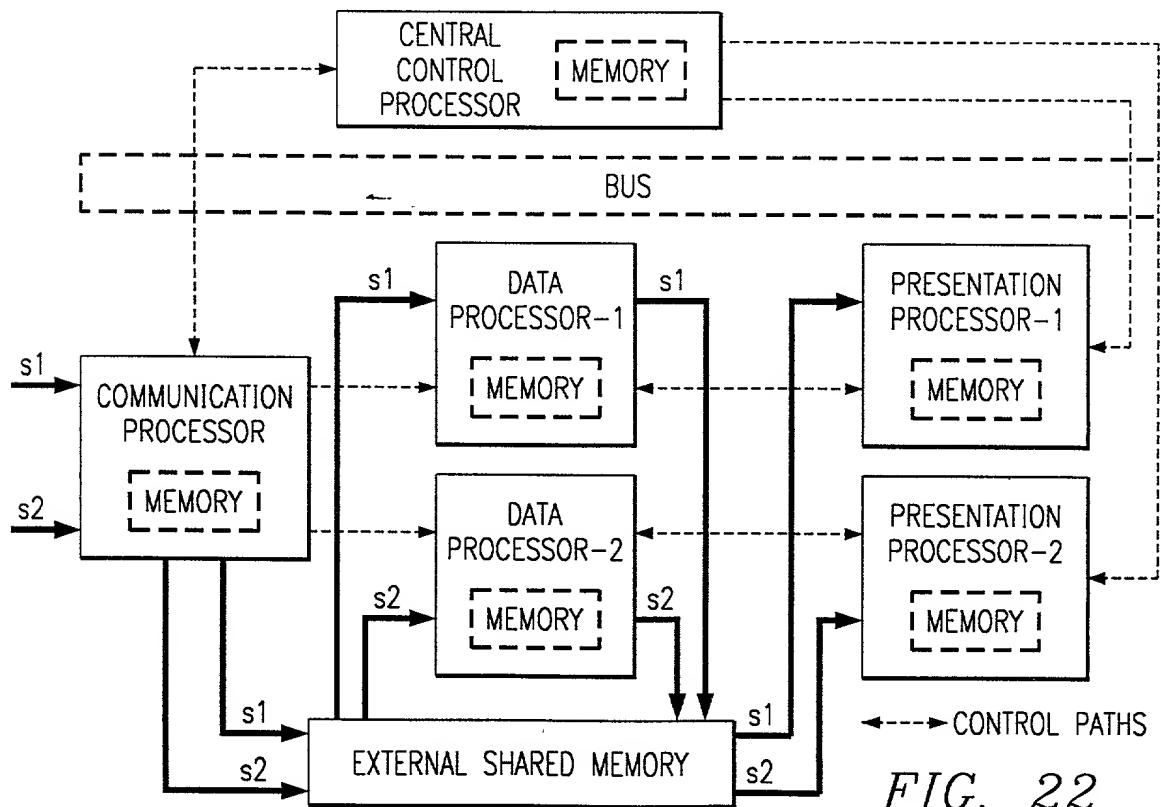


FIG. 23

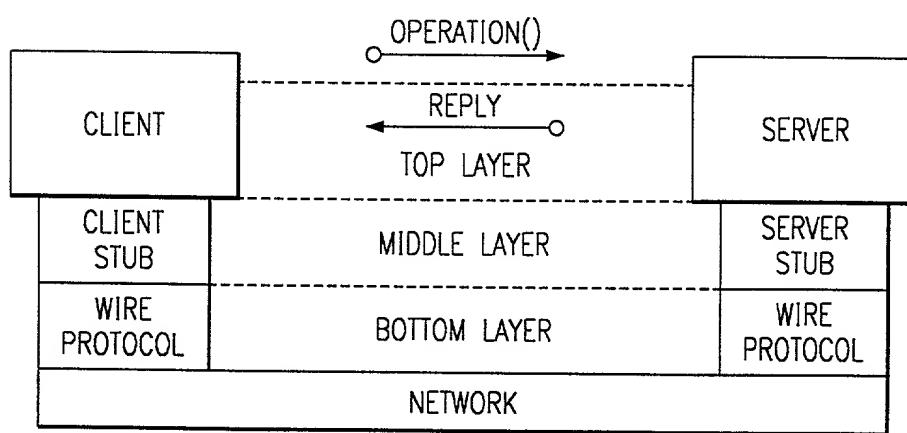
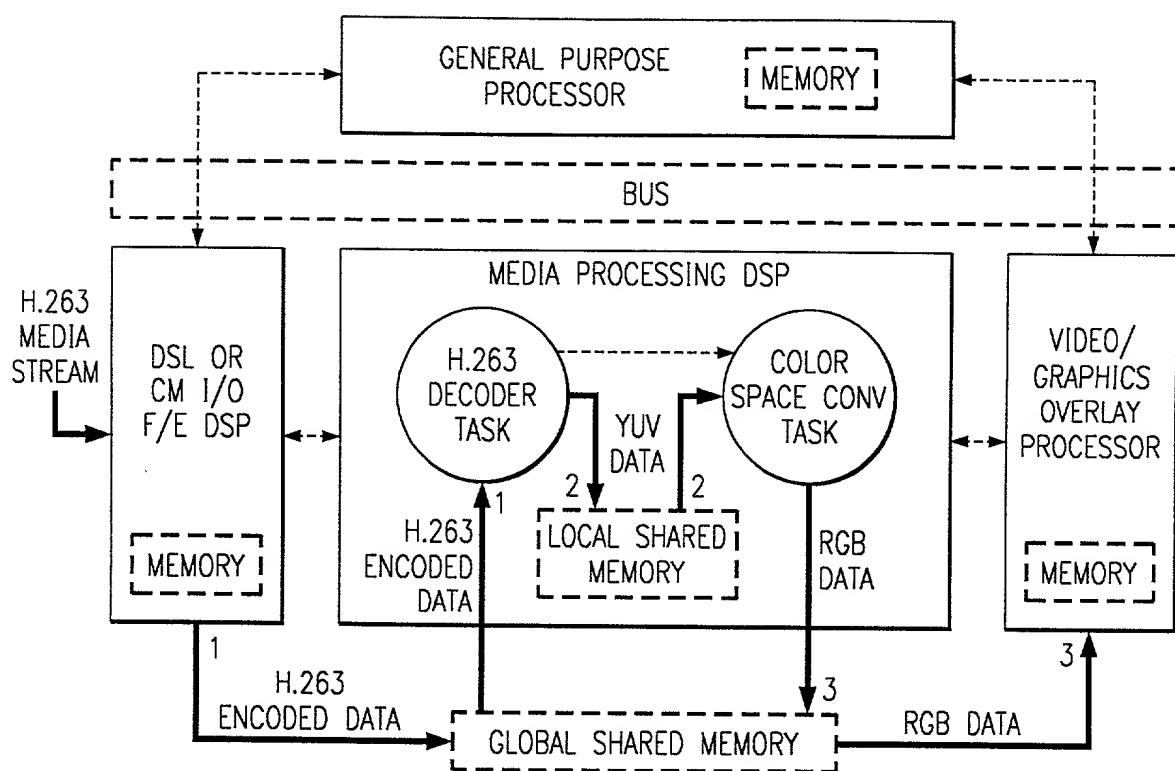


FIG. 24

FIG. 25

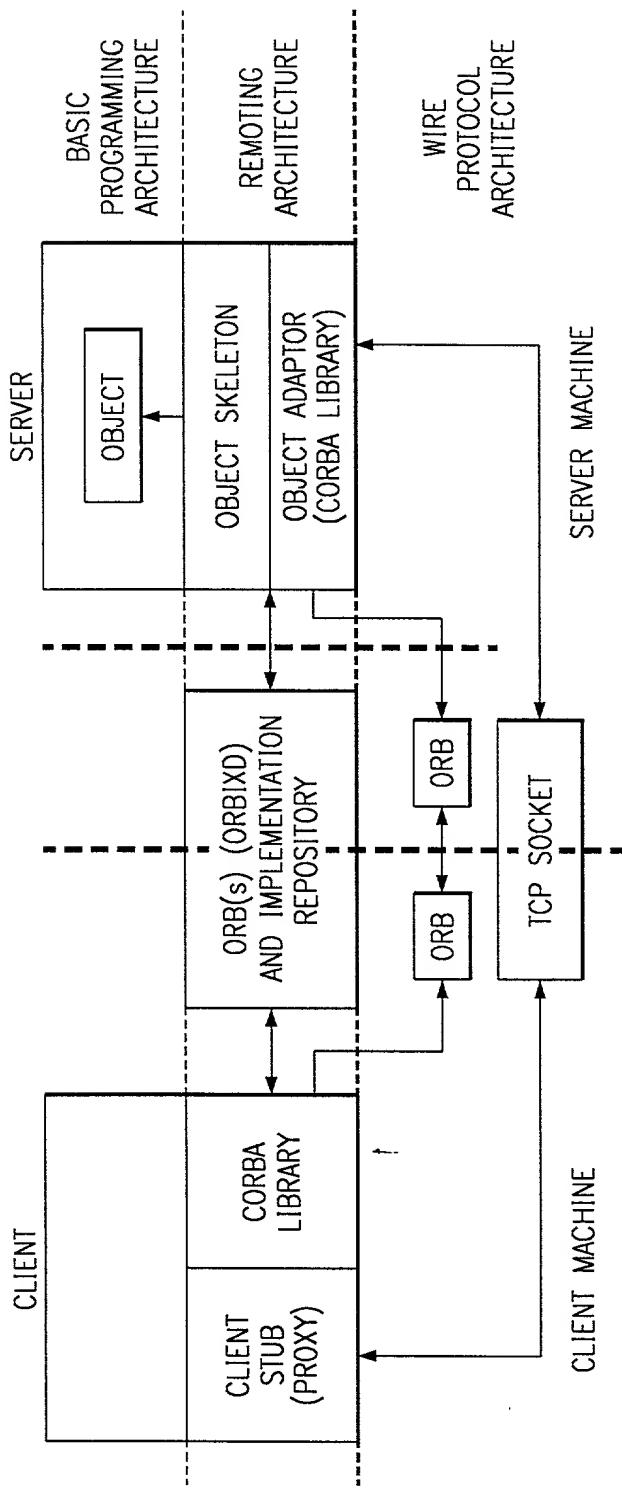
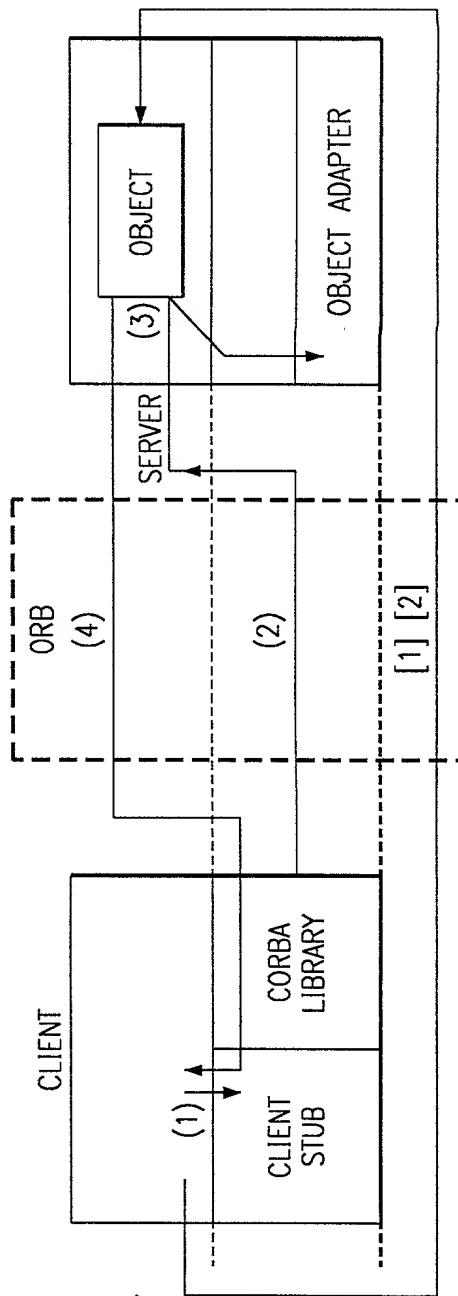


FIG. 26a



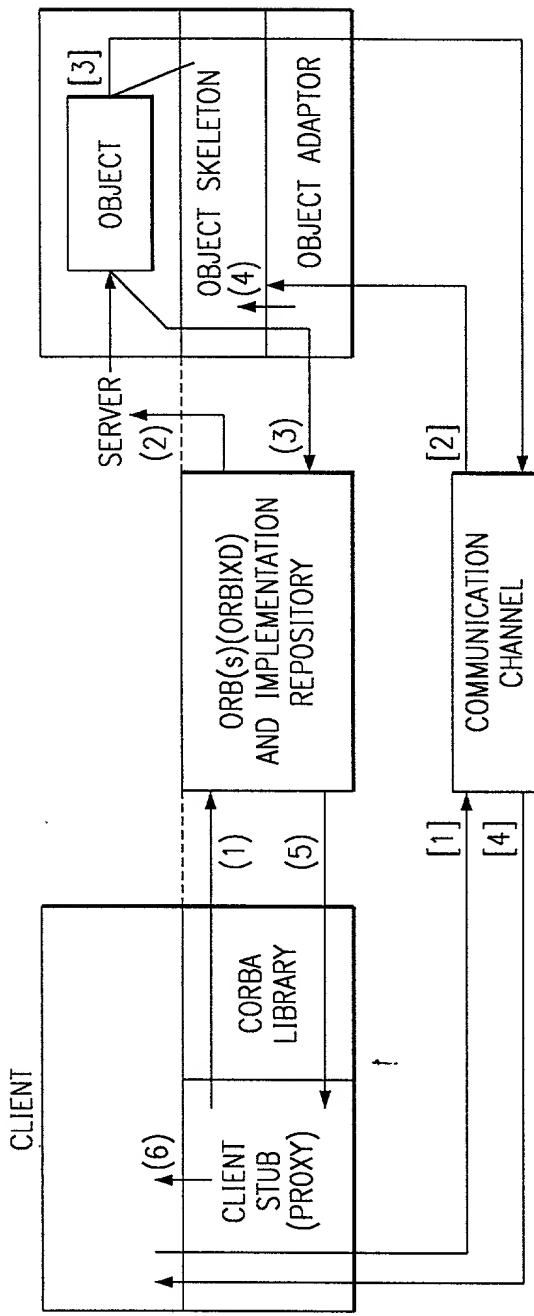
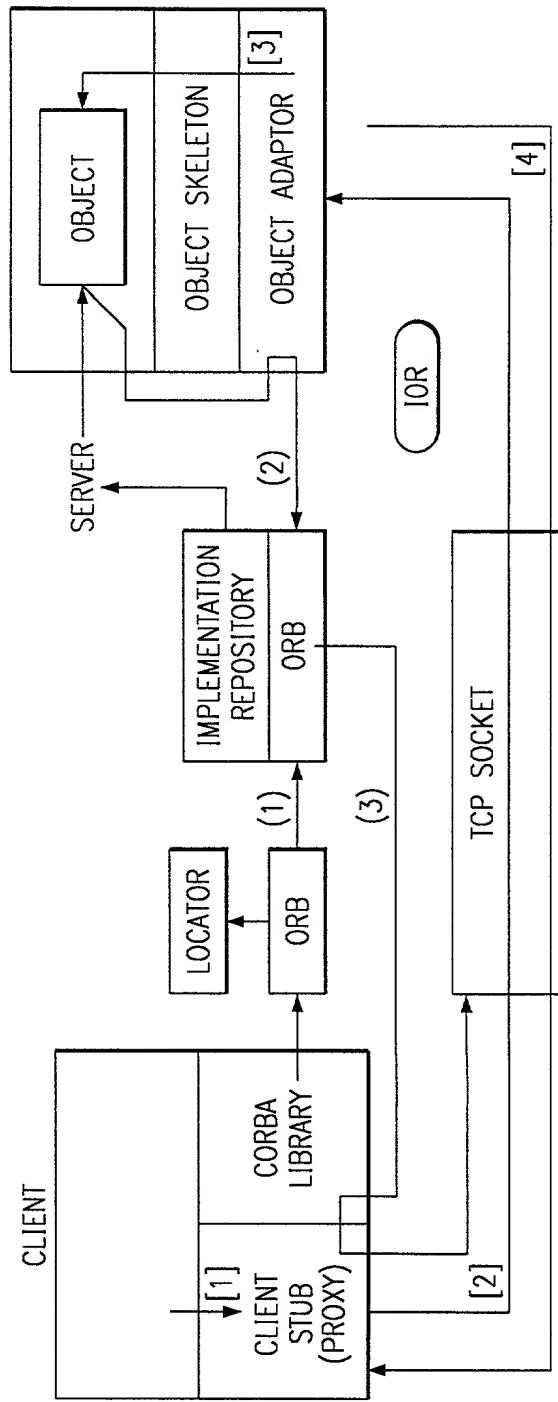


FIG. 26c



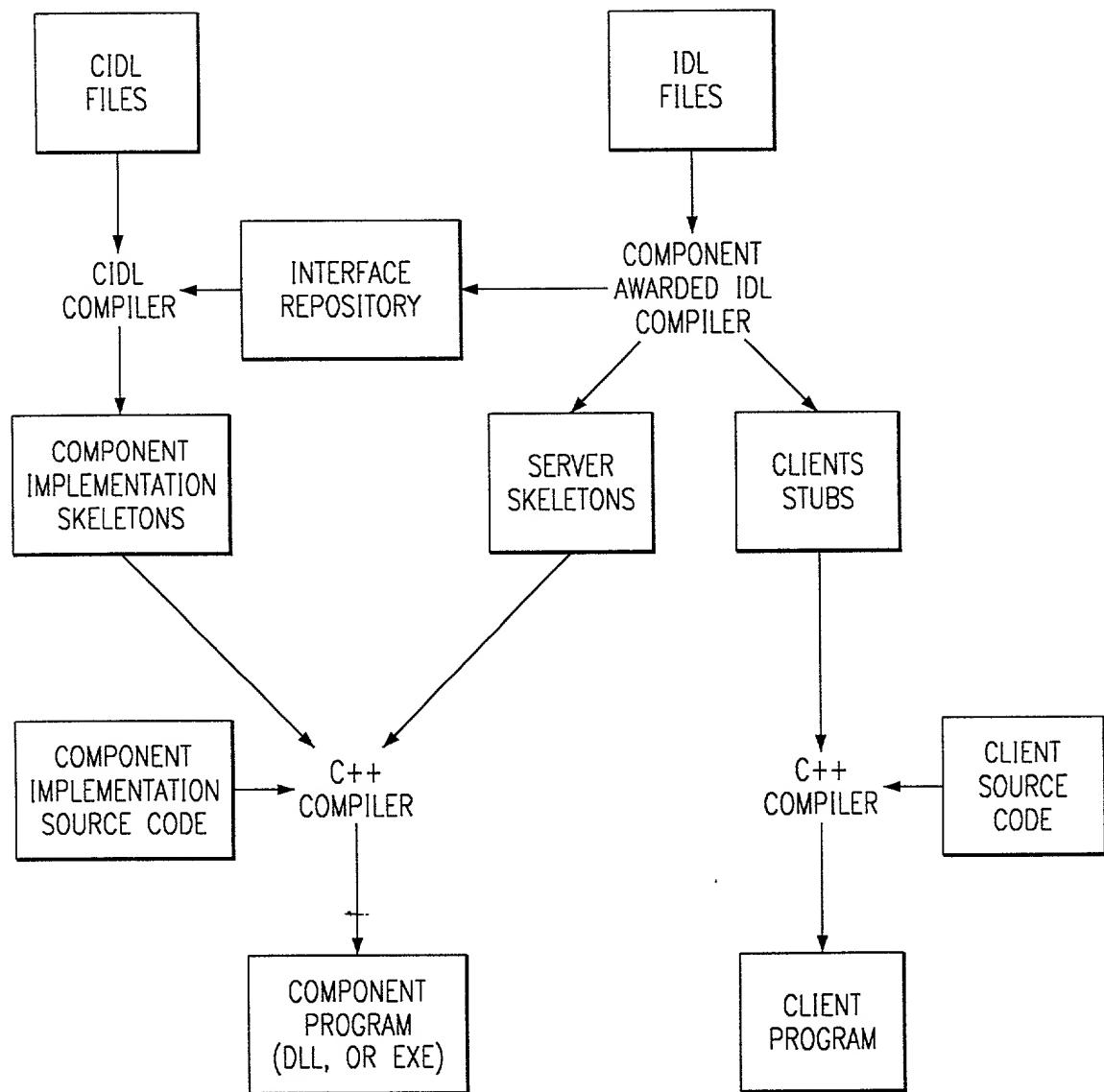


FIG. 27